



**Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет «Высшая школа экономики»**

Факультет компьютерных наук
Департамент программной инженерии

**Рабочая программа дисциплины
Язык Kotlin**

для образовательных программ
«Программная инженерия»,
«Прикладная математика и информатика»
уровень подготовки – **бакалавриат**

Разработчик программы
Мицюк А.А., старший преподаватель, amitsyuk@hse.ru

Одобрена на заседании департамента программной инженерии

«__» _____ 2017 г.

Руководитель департамента

Авдошин С.М. _____

Утверждена Академическим советом образовательной программы «Программная инженерия»

«__» _____ 2017 г., № протокола _____

Академический руководитель образовательной программы

Шилов В.В. _____

Утверждена Академическим советом образовательной программы «Прикладная математика и информатика»

«__» _____ 2017 г., № протокола _____

Академический руководитель образовательной программы

Конушин А.С. _____

Москва, 2017

Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.



1. Область применения

Настоящая программа учебной дисциплины устанавливает требования к образовательным результатам и результатам обучения студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину «Язык Kotlin», студентов и слушателей, желающих принять участие в работе факультатива.

2. Цели освоения дисциплины

В результате освоения учебной дисциплины студенты и слушатели должны овладеть следующими образовательными результатами:

- умение использовать язык программирования Kotlin в профессиональной деятельности для решения практических задач;
- базовое понимание особенностей устройства языка Kotlin, его сходства и отличий с языком программирования Java; понимание особенностей исполнения кода Java-машиной;
- умение самостоятельно находить и усваивать знания, связанные с новыми языками и средами программирования.

3. Приблизительный тематический план учебной дисциплины

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Базовый синтаксис языка. Основные управляющие конструкции.	8	2	2		4
2	Система типов.	8	2	2		4
3	Массивы, диапазоны, строки, коллекции.	6	2	2		2
4	Операторы.	8	2	2		4
5	Объектно-ориентированное программирование с использованием языка Kotlin. Разбор конкретных приёмов и задач.	38	10	10		18
6	Функциональное программирование с использованием языка Kotlin. Разбор конкретных приёмов и задач.	38	10	10		18
7	Инструменты обобщенного программирования в языке.	8	2	2		4
8	Разработка предметно-ориентированных языков с использованием языка Kotlin.	6	1	1		4
9	Использование языка Kotlin совместно с Java.	4	1	1		2
	Итого:	124	32	32		60

Данный тематический план содержит приблизительный список тем и распределение часов. На занятиях (и в задачах) курса возможно также рассмотрение следующих тем:

- Работа с библиотеками для создания графических пользовательских интерфейсов;
- Работа с диском;



- Подготовка тестов для программ на языке Kotlin;
- Экспериментальная функциональность: сопрограммы;
- Разработка с использованием Kotlin мобильных приложений для ОС Android;
- другие темы.

4. Формы контроля знаний студентов

Тип контроля	Форма контроля	Модуль		Параметры
		3-й	4-й	
Текущий	Большое домашнее задание	*		Задача на программирование, выдается примерно на месяц
Итоговый	Экзамен		*	Тест, 80 минут

5. Содержание дисциплины

Раздел 1. Базовый синтаксис языка. Основные управляющие конструкции.

Рассматривается базовый синтаксис языка, ключевые слова, встроенные операторы, основные конструкции потока управления: условные конструкции, циклы.

На практике изучаются основы использования среды разработки IntelliJ IDEA при программировании с использованием языка Kotlin. Ввод кода, подсветка кода, удобные вспомогательные функции: автодополнение и другие, поиск. Работа с проектами.

Обработка исключений в языке Kotlin.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/basic-types.html>
2. <https://kotlinlang.org/docs/reference/control-flow.html>
3. <https://kotlinlang.org/docs/reference/returns.html>
4. <https://kotlinlang.org/docs/reference/packages.html>
5. <https://kotlinlang.org/docs/reference/exceptions.html>

Раздел 2. Система типов.

Рассматриваются особенности системы типов языка Kotlin. Правила приведения (преобразования) типов. Вывод типов в языке Kotlin. Типобезопасность языка Kotlin. Возможности работы с небезопасными типами. Связь с системой типов Java. Особенности использования Java-кода в проектах на Kotlin. Функции расширения.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/typecasts.html>
2. <https://kotlinlang.org/docs/reference/null-safety.html>

Раздел 3. Коллекции.

Массивы. Коллекции. Диапазоны. Работа с Java-коллекциями. Работа со строками как с коллекциями.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/collections.html>
2. <https://kotlinlang.org/docs/reference/ranges.html>
3. <https://kotlinlang.org/docs/reference/java-interop.html>

Раздел 4. Операторы.



Операторы. Перегрузка операторов. Операторы проверки и приведения типов. Элвис-оператор.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/operator-overloading.html>
2. <https://kotlinlang.org/docs/reference/null-safety.html>

Раздел 5. Объектно-ориентированное программирование с использованием языка Kotlin.

Классы, интерфейсы. Методы, поля, свойства (в т.ч. делегируемые). Вложенные классы. Модификаторы доступа. Наследование. Реализация интерфейсов. Конструкторы классов. Расширения. Объекты, объекты-компаньоны. Особенности ООП кратко.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/classes.html>
2. <https://kotlinlang.org/docs/reference/interfaces.html>
3. <https://kotlinlang.org/docs/reference/properties.html>
4. <https://kotlinlang.org/docs/reference/visibility-modifiers.html>
5. <https://kotlinlang.org/docs/reference/nested-classes.html>
6. <https://kotlinlang.org/docs/reference/data-classes.html>
7. <https://kotlinlang.org/docs/reference/enum-classes.html>
8. <https://kotlinlang.org/docs/reference/object-declarations.html>
9. <https://kotlinlang.org/docs/reference/delegation.html>
10. <https://kotlinlang.org/docs/reference/delegated-properties.html>
11. <https://kotlinlang.org/docs/reference/extensions.html>

Раздел 6. Функциональное программирование с использованием языка Kotlin.

Функции в языке Kotlin. Методы. Передача параметров. Лямбда-выражения. Использование лямбда-выражений при работе с коллекциями. Функции высших порядков. Встроенные функции. Особенности функционального стиля кратко.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/functions.html>
2. <https://kotlinlang.org/docs/reference/inline-functions.html>
3. <https://kotlinlang.org/docs/reference/lambdas.html>
4. <https://kotlinlang.org/docs/reference/multi-declarations.html>

Раздел 7. Инструменты обобщенного программирования в языке.

Работа с обобщениями в языке Kotlin. Обобщенные типы. Обобщенные функции. Вариантность. Проекция типов.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/generics.html>

Раздел 8. Разработка предметно-ориентированных языков с использованием языка Kotlin.



Предметно-ориентированные языки (DSL). Встроенные средства Kotlin для разработки предметно-ориентированных языков. Построители.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/type-safe-builders.html>

Раздел 9. Использование языка Kotlin совместно с Java.

Особенности использования языка Kotlin в проектах, содержащих код на языке Java.

Материалы по теме:

1. <https://kotlinlang.org/docs/reference/java-interop.html>
2. <https://kotlinlang.org/docs/reference/java-to-kotlin-interop.html>

Каждый раздел рассматривается на лекциях, а во время семинаров рассмотренный материал используется для решения практических учебных задач.

6. Критерии оценки знаний, навыков

Текущий контроль предусматривает выполнение студентами большого домашнего задания (*Одомашнее задание*) по прошествии примерно половины курса. Домашнее задание состоит в необходимости разработать программу по заданию с использованием возможностей языка Kotlin, рассмотренных на занятиях ранее. В качестве результата студенты сдают исходный код разработанных программ и объясняют, почему их решения устроены тем или иным образом.

Задание выдается студентам преподавателем. В зависимости от количества студентов могут быть предложены разные наборы вариантов задания. Оценка за домашнее задание выставляется с учетом полноты выполнения задания, а также способности студента объяснить примененные при решении приемы.

Также оценивается работа студентов на семинарских занятиях. В ходе занятий студентам предлагаются задания, которые необходимо выполнить с помощью разработки программы на языке Kotlin. Также могут предлагаться тесты, содержащие проверочные вопросы («Какой результат выведет следующая программа?» и подобные). Задачи, которые студент не успел выполнить во время занятия, он может выполнить дома до начала следующего занятия. Работа на семинарах и дома оценивается и учитывается в накопленной оценке (*Оаудиторная*).

На последнем занятии студенты сдают экзамен, который проводится в виде теста (*Оэкзамен*) по всему материалу курса. Длительность экзамена — 80 минут. Проведение экзаменационного тестирования может производиться с помощью электронных тестирующих систем.

7. Образовательные технологии

Студенты изучают особенности языка Kotlin в ходе выполнения заданий по разработке программ на этом языке в аудитории совместно с преподавателем, а также самостоятельно, выполняя домашние задания. Для проверки теоретических основ используются тесты. В рамках курса возможно проведение дополнительных лекций приглашенных докладчиков из числа программистов, использующих язык Kotlin в промышленной разработке.

8. Оценочные средства для текущего контроля и аттестации студента

Примерный перечень вопросов для самопроверки студентов к экзамену:

1. Базовый синтаксис языка. Основные управляющие конструкции.



2. Система типов. Типобезопасность в Kotlin. Правила и механизмы приведения типов.
3. Работа с коллекциями в Kotlin. Диапазоны. Использование Java-коллекций.
4. Операторы. Перегрузка операторов. Операторы проверки и приведения типов. Элвис-оператор.
5. Классы, интерфейсы. Методы, поля, свойства (в т.ч. делегируемые). Вложенные классы. Модификаторы доступа. Наследование. Реализация интерфейсов. Объекты.
6. Функции в языке Kotlin. Передача параметров. Лямбда-выражения. Использование лямбда-выражений при работе с коллекциями. Функции высших порядков.
7. Работа с обобщениями в языке Kotlin. Обобщенные типы. Обобщенные функции. Вариантность. Проекция типов.
8. Предметно-ориентированные языки (DSL). Встроенные средства Kotlin для разработки предметно-ориентированных языков.
9. Использование языка Kotlin в проектах, содержащих код на языке Java: возможности и трудности.

9. Порядок формирования оценок по дисциплине

Накопленная оценка за дисциплину рассчитывается следующим образом:

$$O_{\text{накопленная}} = 0,7 * O_{\text{аудиторная}} + 0,3 * O_{\text{домашнее_задание}}$$

$O_{\text{аудиторная}}$ включает в себя оценки за выполнение аудиторных и домашних работ, предлагаемых на каждом занятии. Эта оценка формируется на основании рабочей ведомости. $O_{\text{домашнее_задание}}$ — оценка за домашнее задание, выполняемое студентами самостоятельно на 4-5 неделях курса.

Результирующая оценка за дисциплину рассчитывается следующим образом:

$$O_{\text{результатирующая}} = 0,8 * O_{\text{накопленная}} + 0,2 * O_{\text{экзамен}}$$

$O_{\text{экзамен}}$ — оценка за финальный тест.

По всем видам работ выставляется 10-балльная оценка. Способ округления — арифметический.

10. Учебно-методическое и информационное обеспечение дисциплины

<https://kotlinlang.org/docs/reference/> - официальный сайт со справочником по языку Kotlin (на английском языке)

Для успешного освоения дисциплины, студент/слушатель использует интегрированную среду разработки JetBrains IntelliJ IDEA версии 2017.1 (или более новую) с подключённым расширением для разработки с использованием языка Kotlin. Может использоваться открытая версия Community Edition, доступная по адресу <https://www.jetbrains.com/idea/>.



11. Материально-техническое обеспечение дисциплины

Занятия проходят в аудитории, оборудованной проектором и «белой» доской. Для выполнения заданий используются компьютеры, на который установлена интегрированная среда разработки JetBrains IntelliJ IDEA версии 2017.1 (или более новая), а также расширение этой среды для разработки с использованием языка Kotlin. Компьютеры подключены к сети Интернет.